The NECC Provisional Technical Transition Architecture Specification

Version 0.5.7

April 12, 2006

1 INTRODUCTION

In September 2005, the Assistant Secretary of Defense (Networks and Information Integration) ASD(NII) designated the Defense Information Systems Agency (DISA) as the lead Agency for material development of a new Joint program tasked to build the next generation of military command and control systems. From the beginning this groundbreaking program—originally known as Joint Command and Control (JC2) but now called Net-Enabled Command Capability (NECC)—was based on a new paradigm. It would not look to the past for its inspiration, a past of large monolithic systems colloquially known as *stove-pipes*, but look forward to the future where capability was distributed across a powerful, flexible, and redundant network. NECC would be based upon a *Net-centric* architecture. It would based on small modular components (i.e., services) organized into Capability Modules that are meant to be agile and adaptable. And it would provide tailored capabilities to the most important NECC customer, the warfighter.

NECC will not be the effort of a single military component but will be a collaboration across the command and control spectrum. DISA will be the lead component[1], managing and directing the contributions of the others, but all the components will participate in shaping the new architecture, especially the military services. Each service will setup its own Component Program Management Office (CPMO) in support of NECC and place that office under the management control of the Joint Program Executive Office (JPEO) established by DISA. The JPEO will have overall responsibility for NECC, to include its technical direction. Joint Forces Command (JFCOM) will be designated as the Operational Sponsor and act as the end-user representative during the design and development of NECC.

1.1 NECC Provisional Architecture

The DOD acquisition process is a formal and highly structured process for managing programs. It is divided into several key phases each with its own milestone. At the end of each phase a program is reviewed to ensure that it has met the goals of that phase. This review is called a *milestone decision* and a successful completion permits the program to advance to the next phase. This process applies to all DOD programs—from tanks to software systems—so it applies to NECC and is the process that NECC must follow.

The first phase in the process is the Technology Development (TD) phase. During this phase, which will last from 18 to 24 months, NECC will investigate and prototype new technologies, ascertain and mitigate the risk of these new technologies, and demonstrate the military utility of these technologies in an operationally relevant environment. The TD phase ends with a program review milestone known as *Milestone B*. The next phase is the System Development and Demonstration (SDD) phase during which NECC will build the actual capabilities planned for employment use by end-users. At the end of the SDD phase, the development effort must pass Milestone C at which point it is approved for operational use by warfighters as the first increment of NECC

[1] Component, in the sense used here, means a military organization, i.e. an agency within the DOD, a major command such as CENTCOM, or one of the military services.

capability. This process of going through phases and milestones is repeated for additional Increments of capability until all user requirements are met.

During the TD phase, which is an experimental phase, NECC will explore the potential of a Net-centric architecture. NECC will test and validate a variety of architectural concepts and patterns in this phase. As part of the collaborative process, components participating in NECC will submit pilot projects for evaluation and analysis. Lessons learned from the TD phase will enable NECC to design an advanced architecture that meets the needs of warfighters.

In January 2006, the JPEO released a white paper entitled *The JC2 Technical Transition Architecture,* which presented a plan for developing the NECC technical architecture using an evolutionary process. The plan proposed that the NECC architecture evolve throughout the TD phase in an incremental and iterative manner. The first step in this process is publishing an initial set of technical guidance, called the Provisional Technical Transition Architecture (PTTA). It is provisional because it is expected that the PTTA will evolve based on the experience and results of the TD phase and that the final architecture, the Technical Transition Architecture (TTA), will not published until the start of the SDD phase.

The NECC PTTA, and the TTA after it, will include a definition of the NECC component layers and their functions, a definition of the interfaces between component layers, a set of initial standards for integration and interoperability across component layers, design guidance, and recommended best practices for NECC developers during the TD phase and, after maturation to the TTA, during the SDD phase.

1.2 NECC PTTA Guiding Principles

Key objectives of the NECC PTTA are to minimize the integration problems and to maximize interoperability among NECC capability modules.

Minimizing the complexity and size of the NECC PTTA will facilitate understanding by the development community.

Specification of a NECC PTTA will help eliminate risk during the TD phase and promote rapid progress to the NECC TTA.

All guidance and standards in the NECC provisional architecture will be consistent with existing DOD policy including guidance and constraints dictated by the Defense Information Technology Standards Registry (DISR).

1.3 Purpose of the Document

The purpose of this document is to describe the initial set of standards and specifications for NECC, and to provide guidance to NECC developers as they build capability modules in the TD phase.

The target audience for this document is systems engineers and software developers, who are involved in the design, engineering, implementation, testing, and operation of NECC services.

NECC capability module developers shall adhere to the standards and guidance defined in this document for all development activities in the NECC TD phase.

2 The NECC Provisional Architecture

2.1 Service Oriented Construct

Over the past several years the Service Oriented Architecture (SOA) concept has emerged as a flexible and scalable approach to distributed information system design and operation. SOAs are based on the concept of *providers* creating and operating *services* (each with well defined public interfaces) for use by authorized *consumers* on the network. The provider is responsible for the engineering and implementation details so that the consumer only needs to understand how to access it and how to interpret the results returned by it (see Figure 2-1). The Global Information Grid (GIG) Net-Centric Enterprise Services (NCES) will be implemented as a SOA.

It is essential to recognize that a SOA is an architectural pattern and does not equate to a specific implementation such as Simple Object Access Protocol (SOAP). SOAP is one mechanism for implementing a SOA, REST[2], for example, is another. Fundamentally, the purpose of a SOA is to isolate the service consumer from internal implementation details. Information is exchanged between a provider and consumer through a limited number of well-defined public interfaces and data schemas.

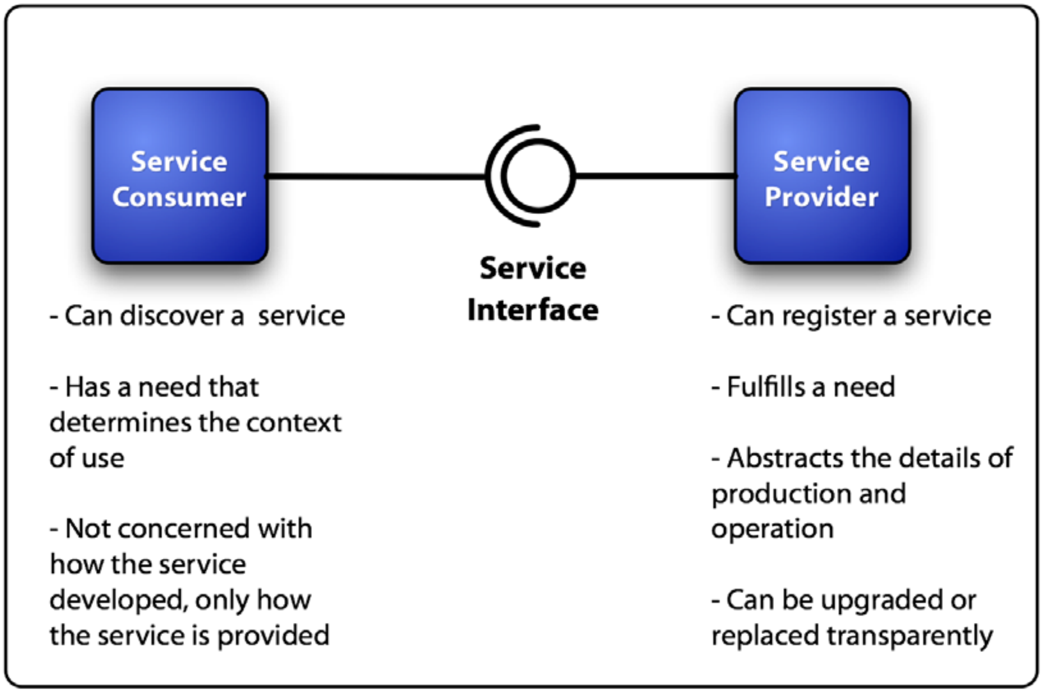
Service Consumer
Service Interface
Service Provider
- Can discover a service
- Has a need that determines the context of use
- Not concerned with how the service developed, only how the service is provided
- Can register a service
- Fulfills a need
- Abstracts the details of production and operation
- Can be upgraded or replaced transparently

Figure 2-1. Service consumers and providers

[2] REST stands for Representational State Transfer but used in this context can be thought of as simply exchanging XML over HTTP without APIs.

An example will illustrate the importance of the SOA concept. In the 1950s the world's cargo transportation industry was a set of discrete domains—shipping, rail, trucking, and air. Starting in the 1960s the cargo transportation industry began to restructure itself in ways that are directly analogous to SOA principles. The key to this restructuring was the adoption of standardized shipping containers as the means for moving cargo and the isolation of the container exchange points (e.g., ship-port, port-rail, rail-truck) to a manageably small number of types of exchange points. The net effect of these changes was to create an *inter-modal* transportation network that provided a huge improvement in speed, efficiency, and cost for customers; to enable new business applications based upon much improved end-to-end transportation capabilities (e.g., just-time supply chains); and to transform the transportation modes into the role of service providers. These changes were achieved not by having to replace the trillions of dollars already invested in the infrastructure of individual transportation modes, but by isolating and integrating the key transportation interface points. The transportation modes are analogous to services in the SOA concept. The reduced number and kind of exchange points are analogous to the well-defined interfaces between providers and consumers in the SOA pattern. Standardized shipping containers are analogous to standardized message formats and protocols, such as Web Services. Finally, the transportation industry restructured itself so that the internal details of its network were hidden. This is analogous to hiding internal implementation details of a service from a consumer.

> " [A service is]… a contractually defined behavior that can be provided by a component for use by any other component, solely based on the interface contract. From the consumer's point of view services are black boxes on the network, in the sense that their internal implementation is hidden. "
>
> - NCES

2.2 The NECC Architecture Foundation

All NECC Architectures will consist of the following three elements:

1. **Technical Interface Specification** A definition of the mechanism and semantics for exchanging information and conducting transactions across a defined NECC architecture boundary. A Technical Interface Specification will be expressed in terms of one or more standards.
2. **Conformance Suite** A software package that provides a means to verify conformance with the standards mandated in a Technical Interface Specification and a reference implementation to address any ambiguities associated with those standards.
3. **Best Practices** A recommended set of practices for operating across NECC architecture boundaries. Best practice constraints are less rigorous than those defined by a Technical Interface Specification, but are valuable in enhancing community-wide activities.

This document will focus on a layered architecture model as a means to define the NECC boundaries. For the sake of clarity and simplicity, a three-layer model is used consisting of a transport layer, an information layer, and an application layer. The application layer includes the session, business, and presentation layers. The three-layer model emphasizes the information layer as a key focus area.

The integration and isolation paradigms of existing C2 architectures and the objective NECC architecture are contrasted in Figure 2-2. Existing C2 architectures were developed largely before the power and potential of networks was as fully developed as it is today. Exercising applications over the network was not feasible. To run an application required that it be loaded directly onto the user's workstation or onto a server at the user's site. Given this environment finding ways to successfully integrate multiple applications within a single platform or operating environment posed a significant challenge.

The NECC architecture is being developed at a time when platform and operating environments are becoming less critical. There is less and less need to consolidate applications on a particular computing platform. Hardware is smaller and less expensive, and services can deliver functionality over much more capable networks. Therefore the NECC architecture can focus on critical integration problems such as creating, deploying, and operating a much more modular, flexible, and interoperable set of C2 capabilities that can be integrated directly into the network. By establishing cleanly isolated layers that define interface points for all major C2 capabilities (e.g., SA, Intel, Log, Fires, etc.) and by establishing interface standards for each layer, NECC will achieve the desired modularity, flexibility, and interoperability. In the NECC architecture, all C2 capabilities will be required to expose a service consistent with the NECC Technical Interface Specification appropriate for its information layer.

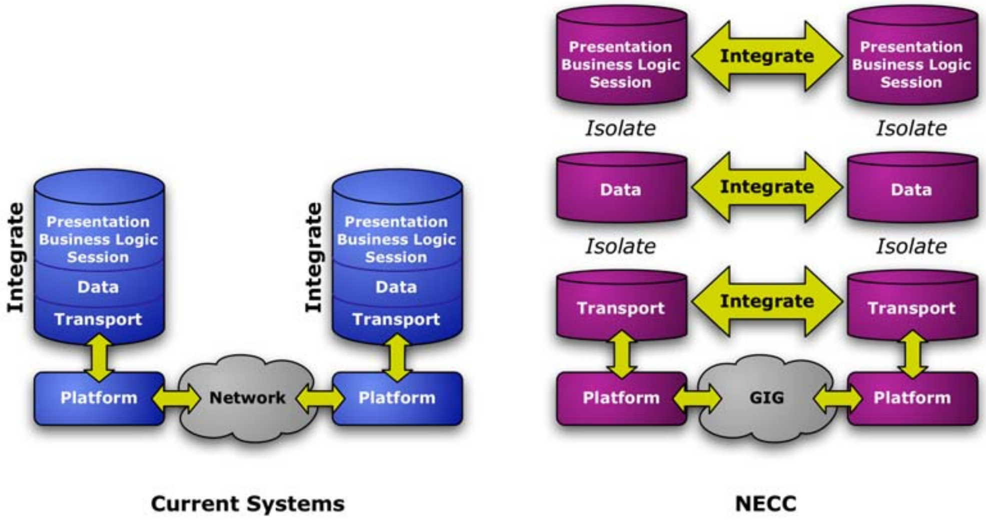

Figure 2-2. Integration Paradigms: Today versus NECC

2.3 Evolving the NECC Architecture

The NECC approach is to evolve from existing architectures into the objective NECC architecture through a series of interim stages, each tied to major NECC increment. This evolutionary concept is depicted in Figure 2-3. The starting point (on the left side of the figure) consists of a system-

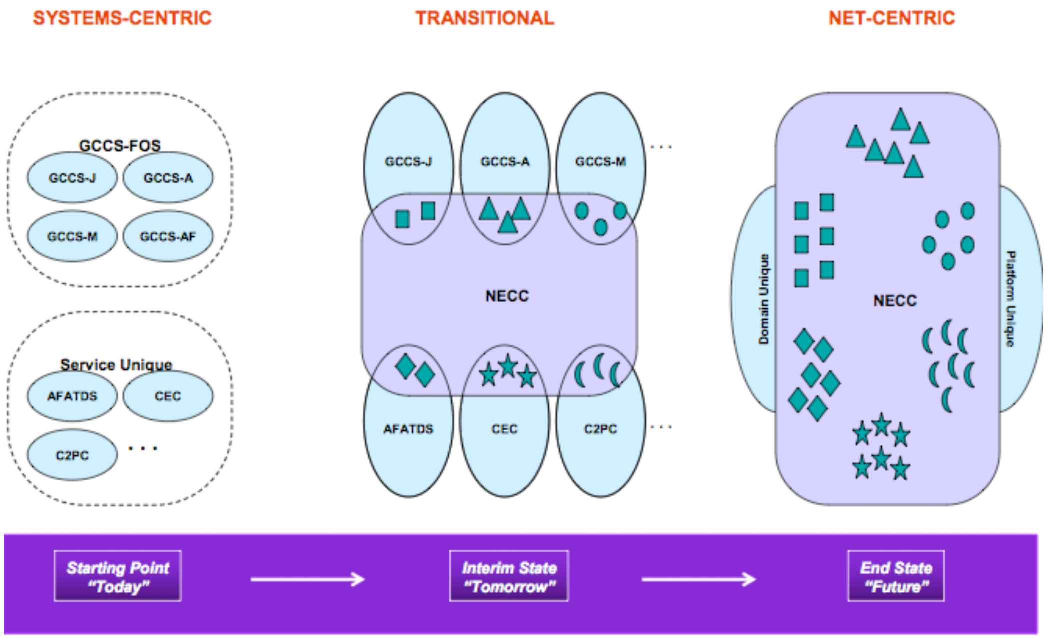

Figure 2-3. Evolving the NECC Architecture

centric state comprised of the current Joint C2 (GCCS[3]) along with a number of service-specific systems that will fall under the NECC umbrella. Today, few of these systems have exposed their capabilities as services. The end-state (depicted on the right side) shows a future environment where all the appropriate capabilities provided by existing systems (indicated by the geometric symbols) are packaged as services on the network. The difficulty in achieving the desired end-state will vary from program to program. To ease the transition to the end-state architecture, the NECC Program will implement an interim state (depicted in the center) as a bridging mechanism. In this interim state, existing systems begin to expose a portion of their C2 capabilities as services, while leaving some of their capabilities to run in a system mode. This transitional state is a hybrid between a services-oriented and system-centric architecture.

3 THE NECC SOFTWARE ARCHITECTURE

3.1 Overview

NECC is a set reusable software services that together form a dynamic and adaptable environment and which can be composed in different ways to provide C2 capability in accordance with the

[3] GCCS stands for Global Command and Control System. It is a Joint system used by all branches of the military and is the predecessor of NECC.

NECC Capability Description Document (CDD). Figure 3-1 (NECC SV-1) shows the top-level NECC hardware and software architecture.

NECC services will run on one of three types of NECC nodes: *fixed*, i.e. located at permanent installations with high performance and high-reliability communications; *transportable*, i.e. packaged to be easily moved from one site and quickly set up at another (but once set up able to count on high-reliability communications); and *mobile*, i.e. placed with or on a moving platform or Command Center.

The primary difference between mobile and fixed nodes is the requirement to function during periods of intermittent or limited communications. Naval vessels, aircraft, and ground operation centers are examples of mobile nodes. For NECC, this means that mobile nodes may require additional capabilities necessary to maintain state during the periods of limited or absent communication with the GIG. Mobile nodes will require some ability to locally host NECC services, which then federate with the larger enterprise, especially with respect to management and caching of relevant data sources and information. This requirement is also relevant to fixed and transportable nodes where communications are reliable but bandwidth is limited.

This document will not define any constraints on hardware, because software architecture and services are the primary focus of the NECC TD phase. However, full architectural specifications for NECC nodes will be included in the final NECC TTA.

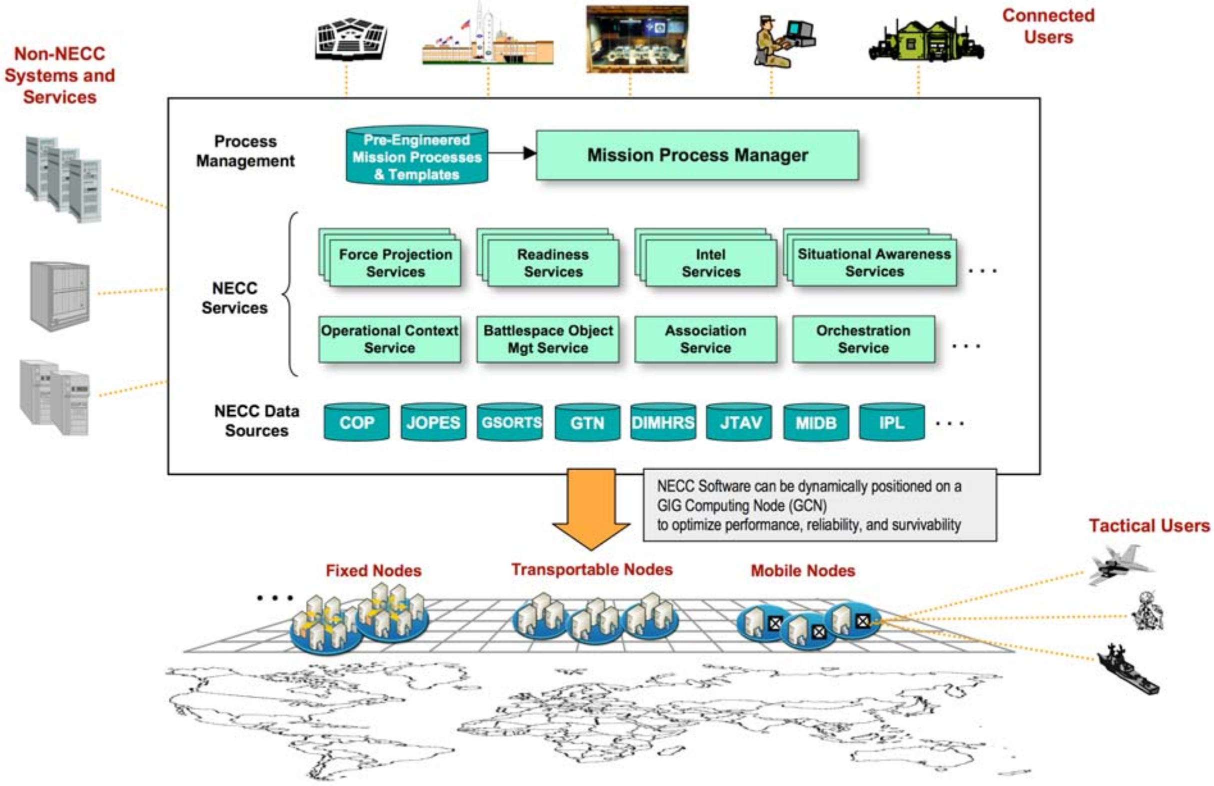
Connected Users
Non-NECC Systems and Services
Process Management
Pre-Engineered Mission Processes & Templates
Mission Process Manager
NECC Services
Force Projection Services
Readiness Services
Intel Services
Situational Awareness Services
. . .
Operational Context Service
Battlespace Object Mgt Service
Association Service
Orchestration Service
. . .
NECC Data Sources
COP
JOPES
GSORTS
GTN
DIMHRS
JTAV
MIDB
IPL
. . .
NECC Software can be dynamically positioned on a GIG Computing Node (GCN) to optimize performance, reliability, and survivability
Tactical Users
Fixed Nodes
Transportable Nodes
Mobile Nodes
. . .

Figure 3-1. NECC SV-1

3.2 Technical Standards for the NECC TD Phase

Throughout the rest of Section 3, the PTTA will specify technical standards for the TD phase. As depicted in Figure 2-2 above, the standards fall into three categories: Transport, Presentation/Business Logic/Session, and Data. Unless noted otherwise, all standards in this section are mandated in the DOD Information Technology Standards Registry (DISR).

Each standards table is divided into three columns: Purpose, Standard, and Guidance. Purpose describes the technical domain of the standard and where it should be used. The Standard column lists the version and name of the relevant specifications. The Guidance column specifies under what circumstances NECC service developers must use the standard.

3.3 Transport Standards

The transport standards for the TD phase are listed in Table 3-1.

Table 3-1. Transport Standards for NECC TD Phase

Purpose	Standard	Guidance
Basic connectivity between the services.	IETF Standard 5, Internet Protocol, September 1981. RFCs 791/950/919/922/792/1112 IETF RFC 2460, Internet Protocol, Version 6 (IPv6) Specification, December 1998 IETF Standard 13/RFC 1034/RFC 1035, Domain Name System, November 1987	IP networking. Accommodate both IPv4 and IPv6 addressing and Network Address Translation. Follow GIG guidelines as they emerge for taking advantage of Quality of Service capabilities of the network.
Service transport protocol	RFC 2246, The Transport Layer Security (TLS) Protocol Version 1.0, January 1999 RFC 2616, Hypertext Transfer Protocol - HTTP 1.1, June 1999 Secure Sockets Layer (SSL) Protocol, Version 3.0, 18 November 1996	HTTPS shall be used as the transport protocol between all service providers and consumers.

3.4 NECC Software Layers

Figure 3-2 (NECC SV-2) breaks down the 3-layer model from Figure 2-2 into 5 layers and illustrates a notional programmatic boundary separating those software elements that are products of the NECC program from those that are products of other programs.

The following sections introduce the software layers and other elements depicted in Figure 3-2. For each layer, the PTTA specifies the Presentation/Business Logic/Session and Data standards that must be used whenever one of the other layers interacts with this layer, and when members of this layer interact with each other.

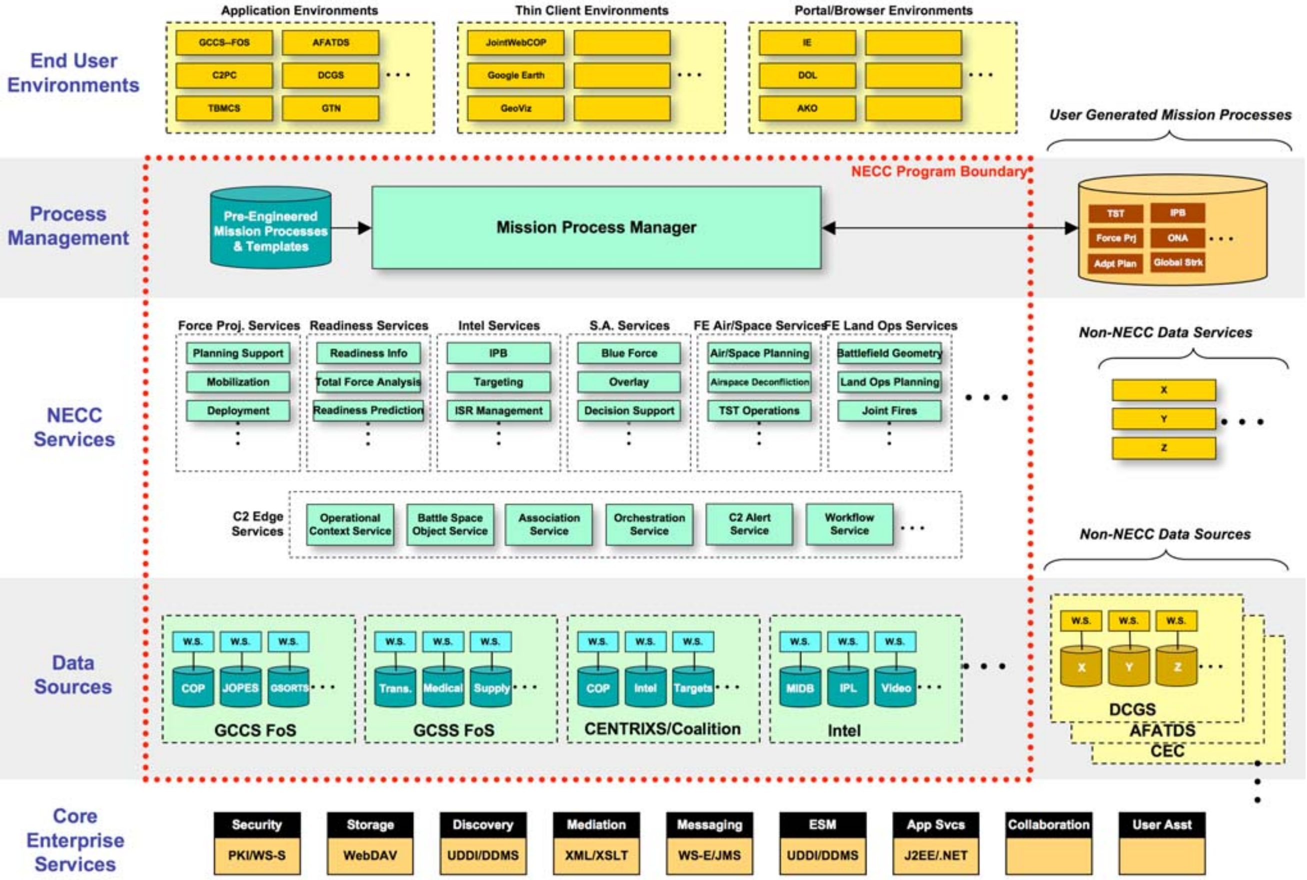

Figure 3-2. NECC SV-2

3.5 Existing Systems

Existing systems and services are included in Figure 3-2. Some of them will within the NECC program boundary and will not. The DCGS FOS, AFATDS, and C2PC[4] are examples of systems outside the NECC program. The Interface Control Documents (ICDs) already in place will govern interaction with such systems during the TD phase. Many of these ICDs are built around complex and extensive military messaging standards, such as the U.S. Message Text Format (USMTF) and the Variable Message Format (VMF). The intent of the NECC technical architecture is to specify the minimum subset of military message formats needed for NECC capabilities. Creating a base-

[4] DCGS, AFATDS, and C2PC are all examples, among the many that could be listed, of existing C2 systems.

line of message formats in use by existing systems and determining which are required for NECC will be accomplished as part of the transition plan for existing systems such as GCCS to NECC. Table 3-2 lists the initial set of military messages that must be supported when NECC capabilities interact with existing systems.

Table 3-2. Military Messaging Information Standards

Purpose	Standard	Guidance
Low bandwidth message exchange.	MIL-STD 6017, Variable Message Format (VMF) K 01.1 – Free Text K 03.6 – Mayday Message K 04.1 – Observation Report K 05.1 – Position Report K 05.14 – SITREP K 05.17 – Overlay K 05.19 – Entity Data	Use for exchanging information with existing systems that process Variable Message Format messages, or for new information exchanges over low bandwidth networks.
General message exchange	MIL-STD 6040, U.S. Message Text Formatting Program A659 (Air Tasking Order) F756 (Air Control Order) D670 (Air SuptReq)	Use for exchanging information with existing systems that process USMTF messages.
General message exchange	OS-OTG (Rev D), Over-The-Horizon Targeting Gold JUNIT	Use for exchanging information with existing systems that process OTH-GOLD messages.
Message exchange over Tactical Data Links	MIL-STD-6016, Tactical Digital Data Link – Joint Tactical Information Distribution System TADIL-J (Link 16)	Use for exchanging information with existing systems that process TADIL-J messages over Link 16.

3.6 Core Enterprise Services

As shown in Figure 3-2, NECC intends to leverage the Core Enterprise Services being provided by the NCES program. NECC developers that are in a position to use these services are encouraged to do so, particularly the Security, Discovery, and Messaging Services. Those that cannot use them in the TD phase should structure their designs to easily adopt them by NECC Milestone B.

The technical standards required for interaction with NCES services are described in the DISA Memorandum, *Proposed Standards for Implementing GIG Enterprise Services*, dated 27 December 2005. The following quoted section of the memo contains the essence of the guidance:

> The following standards and protocols comprise DISA GIG Enterprise Services v1.0:
>
> a. SOAP v1.1 : Simple Object Access Protocol (SOAP) 1.1, W3C Note, 8 May 2000.
>
> b. WSDL v1.1: Web Services Description Language (WSDL) 1.1, W3C Note, 15 March 2001.
>
> c. UDDI v3.0.2: Universal Description, Discovery, and Integration Version 3.0.2 UDDI Spec, 19 October 2004, OASIS Standard, February 2005.
>
> d. W3C XML-Encryption: XML Encryption Syntax and Processing, W3C Recommendation, 10 December 2002.
>
> e. W3C XML-Signature: XML Signature Syntax and Processing, W3C Recommendation, 12 February 2002. DISA Memo, T03, GIG Enterprise Services
>
> f. SAML v1.1: Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) v1.1 , OASIS Standard, 2 September 2003.
>
> g. WS-Security v1.0: Web Services Security: SOAP Message Security 1.0 (WS-Security 2004), OASIS Standard, March 2004.
>
> h. WS-I.ORG Basic Profile v1.1 : Web Services Interoperability Organization (WS-I) Basic Profile 1.1, Final Material, 24 August 2004.
>
> 1) SOAP, WSDL, UDDI
>
> 2) Hypertext Transfer Protocol, HTTP v1.1
>
> 3) RFC2246, TLS Protocol v1.0
>
> 4) RFC2560, x.509 Public Key Infrastructure Certificate
>
> 5) SSL Protocol v3.0
>
> i. XACML v1.0: extensible Access Control Markup Language (XACML) Version 1.0, OASIS Standard, 18 February 2003.
>
> j. XSLT v1.0: XSL Transformations: Version 1.0: W3C Recommendations, 16 November 1999.
>
> k. CSS Level 1: Cascading Style Sheets (CSS) Level 1, W3C Recommendation, 11 January 1999.
>
> l. WebDAV: Data types for Web Distributed Authoring and Versioning (WebDAV) Properties, RFC 4316. Web Distributed Authoring and Versioning.

In addition, NECC capabilities that create products that can be discovered by NCES Content Discovery services must register their metadata in accordance with the DOD Discovery Metadata Specification (DDMS), Version 1.0, 29 September 2003.

3.7 Data Sources and NECC Services

The first layer of the NECC provisional architecture is the collection of network exposed data sources.

The NECC provisional architecture concept is for existing systems to implement PTTA-compliant approaches to enhancing and exposing their information. Access to the data will be through an appropriate set of services that support both read and (where authorized) write capabilities. Using a services-oriented approach will allow NECC to share C2 data across a broad spectrum of domains while at the same time leveraging existing capabilities.

Because NECC data sources will be constructed as network services, the exposed interface will be required to conform to the same standards as NECC services.

NECC data sources and NECC services will also be subject to semantic standards in order to facilitate interoperability. The specific requirements for semantic standards depend on the domain of the data involved and have not yet been established.

The second layer of the NECC provisional architecture is the NECC Services layer. NECC Services are divided into two sub-groups: C2 Edge Services, which represent a set of basic capabilities that are used throughout most, if not all, of the NECC eight Mission Capability Packages functional areas (e.g., Force Projection, Readiness, Intel, etc.), and MCP-specific services, which encapsulate the needs of a specific functional area.

In the provisional architecture NECC services are not responsible for maintaining the authoritative state of any C2 data. That responsibility resides within the NECC Data Sources. However, NECC Services will access (and in some case write back) data from the NECC Data Sources and will then further process the data to provide value-added services to NECC customers.

NECC Services will interact with non-NECC data sources to provide additional information or to provide value-added services that go beyond the specific data managed by the C2 community.

Table 3-3 lists the Presentation/Business Logic/Session standards governing interaction with the NECC data sources and services layers. Table 3-4 lists the corresponding Data standards.

Table 3-3. Presentation/Business Logic/Session Standards for Data Sources and Services

Purpose	Standard	Guidance
Identification and addressing of objects on the network.	RFC 1738, Uniform Resource Locators (URL), 20 December 1994 RFC 2396, Uniform Resource Identifiers (URI), Generic Syntax, August 1998 (updates RFC 1738)	Namespaces within XML documents shall use unique URLs or URIs for the namespace designation.

Purpose	Standard	Guidance
The standardized exchange of SOAP messages with integrity or confidentiality.	Web Services Security: SOAP Message Security 1.0 (WS-Security 2004), OASIS Standard, March 2004	Required for interaction with NCES.
Specific, practical guidance for the development of web services, through constraints and clarifications to their base specifications.	Web Services Interoperability Organization (WS-I) Basic Profile 1.1, Final Material, August 24, 2004 Note that this profile references several other standards associated with web services: *(1) SOAP, WSDL, UDDI* *(2) Hypertext Transfer Protocol, HTTP v1.1* *(3) RFC2246 TLS Protocol v1.0* *(4) RFC2560, x.509 Public Key Infrastructure Certificate*	Conformance to this standards set is required for all SOAP based services.
Aggregating content and interactive web applications from remote sources.	OASIS Web Services for Remote Portlets (WSRP) Specification, v1.0, August 2003	Required for all portlets built for the presentation layer. WSRP shall be used as the cross-platform portlet interoperability interface in lieu of language specific standards such as JSR-168.
Subscribing to or accepting subscriptions for event notification messages.	Web Services Eventing (WS-Eventing). This standard is under development and is not in the DISR.	Required for publish/subscribe based services. Shall be used instead of language dependant standards such as JMS.
Machine to machine messaging using message queues.	Web Services Reliable Messaging, WS-Reliability 1.1.	Required for point-to-point messaging services. Shall be used instead of language dependant standards such as JMS.

Purpose	Standard	Guidance
Accessing geographic data to be displayed as a map or spatially referenced image.	OGC Web Map Service (WMS) Implementation Specification, 1.1.1	All services that publish data which is primarily geospatial and not real-time shall publish their data as a map layer using WMS. Services whose data is primarily geospatial and whose data is real-time shall publish snapshots of the data using WMS.
Custom rendering and filtering extension for WMS.	OGC Styled Layer Description (SLD) Implementation Specification, 1.0.20	All services that publish data using WMS shall support the SLD extensions.
Describing data manipulation operations on geographic features.	OGC Web Feature Service (WFS) Implementation Specification 1.1 (an emerging standard in the DISR)	Required as the service interface for publishing data that is primarily geospatial and which does not require real-time updates.
Electronic interchange of geospatial data as *coverages*, that is, digital geospatial information representing space-varying phenomena.	OGC Web Coverage Service (WCS) Implementation Specification 1.0 (an emerging standard in the DISR)	Required for publishing coverage data.

Table 3-4. Data Standards for Data Sources and Services

Purpose	Standard	Guidance
General formatting of information for sharing or exchange.	Extensible Markup Language (XML), v1.0 3rd Edition XML Schema: Structures 1.0 (XML Schema Part 1) XML Schema: Datatypes 1.0 (XML Schema Part 2) XML Namespaces: W3C (REC-xml-names-19990114)	XML is required for data exchange. XML Schemas and namespaces are required for all XML documents.
Addressing parts of an XML document, designed to be used by both XSLT and XPointer.	XML Path Language Addressing Method (XPath 1.0)	Developer best practice for the implementation of XML based services.
Addressing into the internal structures of XML documents.	XML Pointer Language - Examination and Selection (XPointer 1.0) – not in DISR	Developer best practice for the implementation of XML based services.
Transforming XML documents into other XML documents.	XSL Translation (XSLT 1.0)	Developer best practice for the implementation of XML based services.
Querying XML documents.	XML Query (XQuery) Requirements (an emerging standard in the DISR)	Developer best practice for the implementation of XML based services.
Representing access-control policies.	eXtensible Access Control Markup Language (XACML) Version 2.0, OASIS Standard, 1 February 2005	Required for interaction with NCES.

Purpose	Standard	Guidance
Exchanging information (assertions) for authentication, attributes or authorization	Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.1, OASIS Standard, 2 September 2003	Required for interaction with NCES.
Providing web content or summaries of web content together with links to the full versions of the content, and other metadata.	Really Simple Syndication (RSS) 2.0 Specification – not in DISR	RSS shall be used as the means to publish change notifications for information that is intended for end users. For example, new or changed content on a web portal shall be published using RSS.
Exchanging imagery information	MIL-STD 2500(B) 2, National Imagery Transmission Format (Version 2.1) for the National Imagery Transmission Format Standard, 22 August 1997 with Notice 1, 2 October 1998, and Notice 2, 1 March 2001	Required for the exchange of imagery products except in the presentation services such as WMS.
Representation of a reference frame, reference ellipsoid, fundamental constants, and an Earth Gravitational Model with related geoid.	MIL-STD 2401 DOD World Geodetic System 84 (WGS84), 11 January 1994	All geospatial data served using the WMS, WCS, and WFS standards shall use WGS84.
Accessing or distributing geospatial data using XML.	OGC Geography Markup Language (GML) Encoding Specification, 2.1.1, 1 November 2002	GML shall be used for all geospatial elements within an XML document.

Purpose	Standard	Guidance
Processing foreign language character sets.	ISO/IEC 10646-1: 2000 Universal Multiple-Octet Coded Character Set (UCS), Part 1: Architecture and Basic Multilingual Plane	All XML documents must use Unicode (UTF-8 or UTF-16) encoding.

3.8 NECC Process Management

To support specific warfighting requirements that involve the coordination of activities among multiple functional areas within the C2 domain, NECC will include a process management layer.

The Process Management layer will provide the mechanism for sequencing a set of NECC services to achieve a specific warfighting process. A set of sequenced NECC services can also be referred to as a Mission Thread.

Mission Threads can be created in two ways: either by developers and stored as templates, or by warfighters in response to a new operational need. The process of dynamically constructing a mission thread by a user is referred to as orchestration.

Table 3-5 lists the required Information standards for interacting with this layer.

Table 3-5. Information Standards for Process Management

Purpose	Standard	Guidance
Defining abstract and executable business processes.	Business Process Execution Language (BPEL4WS) for Web Services, Version 1.1 – not in DISR	Required as the interface for cross-service business process orchestration.

3.9 End User Environments

It is the intent of the NECC provisional architecture to minimize the impact of NECC capabilities on the end user environment (EUE). In many cases NECC will be able to deliver its capabilities without requiring the installation of NECC-specific software on the EUE. For example, many NECC capabilities will be accessible via web pages or via browser plug-ins.

In the NECC provisional architecture there will be three classes of EUE: (1) Application Environments (e.g., TBMCS, C2PC, etc.)[5]; (2) Portal Environments (e.g., DOL, AKO, etc.)[6]; and (3) Brows-

[5] These are typically stand-alone applications (also known as thick clients.)

[6] DOL (Defense On Line) and AKO (Army Knowledge Online) are examples of existing military portals.

Browsers. Application EUEs will interact with the NECC services via the web-service interface. Portals and browsers will require NECC services to provide a presentation capability using web pages.

To ensure interoperability between NECC Services and the EUE, NECC will establish the interface standards between EUEs and NECC services. These standards will include protocols, as well as minimum CPU and memory requirements. Customers will be responsible to ensure their EUE is configured to work with NECC provided services; although ultimately it is the responsibility of NECC to provide (through whatever partnership mechanisms are appropriate) a visualization and end-user experience which meets CDD requirement.

Table 3-6 lists the required Information standards for interacting with this layer.

Table 3-6. Information Standards for End User Environments

Purpose	Standard	Guidance
Displaying content within web browsers.	W3C Hypertext Markup Language HTML 4.0.1 W3C Extensible Hypertext Markup Language XHTML 1.0 W3C Cascading Style Sheets CSS 2.0	NECC applications must support the following browsers: Microsoft Internet Explorer v6.0 and newer, and Mozilla Firefox 1.0 and newer. When a supported browser is not true to the standard, choose to support the browser
Browser plug-ins.	Browser plug-ins are not covered by a single specification. Plug-ins must adhere to DOD Mobile Code policy guidance of Nov 7, 2000.	The use of ActiveX controls in the browser is not allowed.
Display of military symbology.	MIL-STD 2525B(1), Common Warfighting Symbology, 1 July 2005 with Notice of Change 1	All presentation service shall render tracks, tactical graphics, and MOOTW objects using this standard except in the case where the object being rendered are not covered in the standard (e.g. powerlines, etc.)

4 ADDITIONAL GUIDANCE

A driving NECC objective is the development of components that expose their capabilities via the network, eliminating the need for service providers to conform to a specific hardware environment. On the other hand, deployment of NECC services and capabilities in transportable and mobile nodes will require hosting NECC software on the unique hardware deployed at those nodes. Therefore, while NECC Service developers should feel free to select the hardware and operating system environment most appropriate for their services—provided the selection can be successfully accredited (see additional security guidance in section 4.2)—these implementations must be hardware and operating system independent.

NECC developers are strongly encouraged to follow the Net-Centric Enterprise Solutions for Interoperability (NESI)[7] guidance as a model for the development of services.

4.1 Network and Security Guidance

During the NECC TD phase, prototyping will take place on the NIPRNET and SIPRNET, therefore NECC developers must assume network behavior and performance consistent with the existing limits of these networks. Not all services should be developed assuming high bandwidth and reliable networks.

NECC will not be required to operate over non-IP networks; therefore all NECC component developers shall build their components assuming an IP environment.

NECC services must be able to function in a network environment containing firewalls and various routing and filtering schemes, therefore, NECC developers must use standard and well-known ports wherever possible, and document non-standard ports as part of their service interface.

During the TD phase, NECC components will have to undergo security accreditation; therefore component developers must implement their services in accordance with appropriate NECC security guidelines[8].

4.2 Performance and Scalability Guidance

One of the biggest issues that the NECC SOA must address is performance. NECC service developers will be required to provide performance metrics for their services, particularly those services that use the request-response design pattern. Services which support end users must meet the most strict performance measures, which may be as little as 1–5 seconds, the amount a user is willing to tolerate.

NECC service providers shall quantify the performance of their service in terms of the maximum time to return a request.

NECC service providers shall rate their service for the maximum number of consumers it can support while still being able to meet the maximum response time.

For the TD phase it is not necessary to provide services and components that operate on the scale that will be necessary for the SDD phase. However, it will be important to demonstrate that the service will be able to scale. Developers should consider implementation strategies for their services that make it easy to scale, particularly if they are providing capabilities expected to be included in the first NECC Baseline.

[7] Information about NESI can be found at http://nesipublic.spawar.navy.mil/.

[8] These security guidelines are currently under development.

4.3 Recommended Design Patterns

Within the NECC provisional architecture, services must be designed around the Request/Response, Publish/Subscribe, or Message Queue patterns.

For the provisional architecture, developers must provide read or read/write services as appropriate.

For the provisional architecture, developers must implement either synchronous or asynchronous services

For the provisional architecture, developers must include authentication as part of their service.

For the provisional architecture, developers should strive to make their services as granular as possible to facilitate efficiency, modular design, and loose coupling among services.

For the provisional architecture, developers must support dynamic bindings.

5 THE WAY FORWARD

The guidance and standards specified in this document are intended to provide initial guidance for NECC material developers so that they can begin to support the TD Phase. Additional more detailed information about the NECC PTTA will be released over the next several months including:

1. **Development Guidance** The NECC JPMO will write a Developer Guidance document that contains more developer-level detail than this document contains. It will draw heavily from the guidance put forth in NESI.
2. **Transition Guidance** The GCCS PMO will produce a document that provides guidance for transitioning existing GCCS Family-of-Systems capabilities to the NECC PTTA. The document, currently near completion, will be released under the title *GCCS FOS Functionality Transition Plan.*
3. **Embedded Training Guidance** The NECC Joint Program Management Office (JPMO), in conjunction with USJFCOM will publish a document defining the requirements and style guidance for embedded help and training in NECC capabilities. All NECC capabilities will be required to have some level of embedded help and training.
4. **Certification Guidance** The NECC JPMO will provide a Certification Guidance document that delineates the procedures for getting NECC Capabilities certified to operate on the GIG. NECC capabilities will be required to undergo a series of progressively more rigorous certifications to become approved for operation on the GIG. Becoming certified to operate requires demonstrating technical maturity, operational relevance, operational utility, net-readiness, and security accreditation. The NECC program is establishing the Federated Development and Certification Environment (FDCE) for managing this certification process.
5. **Piloting Guidance** The NECC JPMO will develop a Piloting Guidance document that defines the piloting events that will be used by NECC to help mature NECC capabilities. This document will describe the criteria for participation in NECC piloting events.

6. **Operations and Sustainment Guidance** NECC capabilities that become operational must be sustained. The NECC JPMO will produce an Operations and Sustainment document that defines sustainment requirements for NECC Service Providers.